GNU Info

for version 4.5, 6 November 2002

Brian J. Fox (bfox@gnu.org)

This manual is for GNU Info (version 4.5, 6 November 2002), a program for viewing documents in Info format (usually created from Texinfo source files).

Copyright © 1992, 1993, 1996, 1997, 1998, 1999, 2001, 2002 Free Software Foundation, Inc.

> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with the Front-Cover texts being "A GNU Manual," and with the Back-Cover Texts as in (a) below. A copy of the license is included in the section entitled "GNU Free Documentation License."
>
> (a) The FSF's Back-Cover Text is: "You have freedom to copy and modify this GNU Manual, like GNU software. Copies published by the Free Software Foundation raise funds for GNU development."

Table of Contents

1 What is Info? 1

2 Invoking Info 2

3 Moving the Cursor 6

4 Moving Text Within a Window 8

5 Selecting a Node 11

6 Searching an Info File 14

7 Selecting Cross References 16
- 7.1 Parts of an Xref 16
- 7.2 Selecting Xrefs 16

8 Manipulating Multiple Windows 18
- 8.1 The Mode Line 18
- 8.2 Window Commands 18
- 8.3 The Echo Area 19

9 Printing Nodes 23

10 Miscellaneous Commands 24

11 Manipulating Variables 26

12 Customizing Key Bindings and Variables . . 29
- 12.1 Invoking `infokey` 29
- 12.2 `infokey` source format 29

Appendix A Copying This Manual 32
- A.1 GNU Free Documentation License 32
  - A.1.1 ADDENDUM: How to use this License for your documents 38

Appendix B Index 39

1 What is Info?

Info is a program which is used to view Info files on an ASCII terminal. *Info files* are the result of processing Texinfo files with the program `makeinfo` or with one of the Emacs commands, such as `M-x texinfo-format-buffer`. Texinfo itself is a documentation system that uses a single source file to produce both on-line information and printed output. You can typeset and print the files that you read in Info.

2 Invoking Info

GNU Info accepts several options to control the initial node being viewed, and to specify which directories to search for Info files. Here is a template showing an invocation of GNU Info from the shell:

```
info [option]... [menu-item...]
```

The program accepts the following options:

`--apropos=`*`string`*
: Specify a string to search in every index of every Info file installed on your system. Info looks up the named *string* in all the indices it can find, prints the results to standard output, and then exits. If you are not sure which Info file explains certain issues, this option is your friend. Note that if your system has a lot of Info files installed, searching all of them might take some time.

  You can invoke the apropos command from inside Info; see Chapter 6 [Searching Commands], page 14.

`--directory` *`directory-path`*
`-d` *`directory-path`*
: Prepend *directory-path* to the list of directory paths searched when Info needs to find a file. You may issue `--directory` multiple times; once for each directory which contains Info files. The list of directories searched by Info is constructed from the value of the environment variable `INFOPATH`; `--directory` causes the named *directory-path* to be prepended to that list. The value of `INFOPATH` is a list of directories usually separated by a colon; on MS-DOS/MS-Windows systems, the semicolon is used. If you do not define `INFOPATH`, Info uses a default path defined when Info was built as the initial list of directories. If the value of `INFOPATH` ends with a colon (or semicolon on MS-DOS/MS-Windows), the initial list of directories is constructed by appending the build-time default to the value of `INFOPATH`.

`--dribble=`*`dribble-file`*
: Specify a file where all user keystrokes will be recorded. This file can be used later to replay the same sequence of commands, see the '`--restore`' option below.

`--file` *`filename`*
`-f` *`filename`*
: Specify a particular Info file to visit. By default, Info visits the file `dir`; if you use this option, Info will start with `(`*`filename`*`)Top` as the first file and node.

  If *filename* is an absolute file name, or begins with '`./`' or '`../`', Info looks for *filename* only in the directory of the specified *filename*, and adds the directory of *filename* to the value of `INFOPATH`. In contrast, if *filename* is in the form of a relative file name, but without the '`./`' or '`../`' prefix, Info will only look for it in the directories specified in `INFOPATH`. In other words, Info does *not* treat file names which lack '`./`' and '`../`' prefix as relative to the current directory.

In every directory Info tries, if *filename* is not found, Info looks for it with a number of known extensions of Info files[1]. For every known extension, Info looks for a compressed file, if a regular file isn't found. Info supports files compressed with `gzip`, `bzip2`, `compress` and `yabba` programs; it calls `gunzip`, `bunzip2`, `uncompress` and `unyabba`, accordingly, to decompress such files. Compressed Info files are assumed to have '`.z`', '`.gz`', '`.bz2`', '`.Z`', or '`.Y`' extensions, possibly in addition to one of the known Info files extensions[2].

`--help`
`-h`
: Produces a relatively brief description of the available Info options.

`--index-search` *string*
: After processing all command-line arguments, go to the index in the Info file and search for index entries which matche *string*. If such an entry is found, the Info session begins with displaying the node pointed to by the first matching index entry; press `,` to step through the rest of the matching entries. If no such entry exists, print '`no entries found`' and exit with nonzero status. This can be used from another program as a way to provide online help, or as a quick way of starting to read an Info file at a certain node when you don't know the exact name of that node.

  This command can also be invoked from inside Info; see Chapter 6 [Searching Commands], page 14.

`--node` *nodename*
`-n` *nodename*
: Specify a particular node to visit in the initial file that Info loads. This is especially useful in conjunction with `--file`[3]. You may specify `--node` multiple times; for an interactive Info, each *nodename* is visited in its own window, for a non-interactive Info (such as when `--output` is given) each *nodename* is processed sequentially.

`--output` *filename*
`-o` *filename*
: Specify *filename* as the name of a file to which to direct output. Each node that Info visits will be output to *filename* instead of interactively viewed. A value of `-` for *filename* specifies the standard output.

`--raw-escapes`
`-R`
: Do not remove ANSI escape sequences from man pages. Some versions of Groff, the GNU document formatter, produce man pages with ANSI escape sequences for bold, italics, and underlined characters, and for colorized text. By default, Info removes those escape sequences before it displays the man page. If your

---

[1] '`.info`', '`-info`', '`/index`', and '`.inf`'.

[2] The MS-DOS version allows for the Info extension, such as `.inf`, and the short compressed file extensions, such as '`.z`' and '`.gz`', to be merged into a single extension, since DOS doesn't allow more than a single dot in the basename of a file. Thus, on MS-DOS, if Info looks for '`bison`', file names like '`bison.igz`' and '`bison.inz`' will be found and decompressed by `gunzip`.

[3] Of course, you can specify both the file and node in a `--node` command; but don't forget to escape the open and close parentheses and whitespace from the shell as in: `info --node "(emacs)Buffers"`.

terminal supports these escapes, use `--raw-escapes` to let the terminal handle them and display the man pages with those attributes.

`--restore=`*`dribble-file`*
: Read keystrokes from *dribble-file*, presumably recorded during previous Info session (see the description of the '`--dribble`' option above). When the keystrokes in the files are all read, Info reverts its input to the usual interactive operation.

`--show-options`
`--usage`
`-O`
: This option causes Info to look for the node that describes how to invoke the program and its command-line options, and begin the session by displaying that node. It is provided to make it easier to find the most important usage information in a manual without the need to wade through complex menu hierarchies. The effect is similar to the `M-x goto-invocation` command (see [goto-invocation], page 12) from inside Info.

`--speech-friendly`
`-b`
: On MS-DOS/MS-Windows only, this option causes Info to use standard file I/O functions for screen writes. (By default, Info uses direct writes to the video memory on these systems, for faster operation and colored display support.) This allows the speech synthesizers used by blind persons to catch the output and convert it to audible speech.

`--subnodes`
: This option only has meaning when given in conjunction with `--output`. It means to recursively output the nodes appearing in the menus of each node being output. Menu items which resolve to external Info files are not output, and neither are menu items which are members of an index. Each node is only output once.

`--version`
: Prints the version information of Info and exits.

`--vi-keys`
: This option binds functions to keys differently, to emulate the key bindings of `vi` and Less. The default key bindings are generally modeled after Emacs. (See Chapter 12 [Custom Key Bindings], page 29, for a more general way of altering GNU Info's key bindings.)

`menu-item`
: Info treats its remaining arguments as the names of menu items. The first argument is a menu item in the initial node visited (generally `dir`), the second argument is a menu item in the first argument's node, etc. You can easily move to the node of your choice by specifying the menu names which describe the path to that node. For example,

```
info emacs buffers
```

  first selects the menu item '`Emacs`' in the node '`(dir)Top`', and then selects the menu item '`Buffers`' in the node '`(emacs)Top`'.

To avoid searching the '`dir`' files and just show some arbitrary file, use '`-f`' and the filename, as in '`info -f ./foo.info`'.

The index search and the search for the node which describes program invocation and command-line options begins *after* processing all the command-line menu items. Therefore, the Info file searched for the index or the invocation node is the file where Info finds itself after following all the menu items given on the command line. This is so '`info emacs --show-options`' does what you'd expect.

3 Moving the Cursor

Many people find that reading screens of text page by page is made easier when one is able to indicate particular pieces of text with some kind of pointing device. Since this is the case, GNU Info (both the Emacs and standalone versions) have several commands which allow you to move the cursor about the screen. The notation used in this manual to describe keystrokes is identical to the notation used within the Emacs manual, and the GNU Readline manual. See section "Character Conventions" in *the GNU Emacs Manual*, if you are unfamiliar with the notation[1].

The following table lists the basic cursor movement commands in Info. Each entry consists of the key sequence you should type to execute the cursor movement, the `M-x`[2] command name (displayed in parentheses), and a short description of what the command does. All of the cursor motion commands can take a *numeric* argument (see Chapter 10 [to find out how to supply them], page 24. With a numeric argument, the motion commands are simply executed that many times; for example, a numeric argument of 4 given to `next-line` causes the cursor to move down 4 lines. With a negative numeric argument, the motion is reversed; an argument of -4 given to the `next-line` command would cause the cursor to move *up* 4 lines.

⟨C-n⟩ (`next-line`)
⟨DOWN⟩ (an arrow key)
: Move the cursor down to the next line.

⟨C-p⟩ (`prev-line`)
⟨UP⟩ (an arrow key)
: Move the cursor up to the previous line.

⟨C-a⟩ (`beginning-of-line`)
⟨Home⟩ (on DOS/Windows only)
: Move the cursor to the start of the current line.

⟨C-e⟩ (`end-of-line`)
⟨End⟩ (on DOS/Windows only)
: Move the cursor to the end of the current line.

⟨C-f⟩ (`forward-char`)
⟨RIGHT⟩ (an arrow key)
: Move the cursor forward a character.

⟨C-b⟩ (`backward-char`)
⟨LEFT⟩ (an arrow key)
: Move the cursor backward a character.

⟨M-f⟩ (`forward-word`)
C-⟨RIGHT⟩ (on DOS/Windows only)
: Move the cursor forward a word.

---

[1] Here's a short summary. *C-x* means press the *CTRL* key and the key *x*. *M-x* means press the *META* key and the key *x*. On many terminals th *META* key is known as the *ALT* key. *SPC* is the space bar. The other keys are usually called by the names imprinted on them.

[2] `M-x` is also a command; it invokes `execute-extended-command`. See section "Executing an extended command" in *the GNU Emacs Manual*, for more detailed information.

⟨M-b⟩ (`backward-word`)
`C-`⟨LEFT⟩ (on DOS/Windows only)
: Move the cursor backward a word.

⟨M-<⟩ (`beginning-of-node`)
⟨C-⟨Home⟩⟩ (on DOS/Windows only)
⟨b⟩
⟨M-b⟩, vi-like operation
: Move the cursor to the start of the current node.

⟨M->⟩ (`end-of-node`)
⟨C-⟨End⟩⟩ (on DOS/Windows only)
⟨e⟩ : Move the cursor to the end of the current node.

⟨M-r⟩ (`move-to-window-line`)
: Move the cursor to a specific line of the window. Without a numeric argument, `M-r` moves the cursor to the start of the line in the center of the window. With a numeric argument of *n*, `M-r` moves the cursor to the start of the *n*th line in the window.

4 Moving Text Within a Window

Sometimes you are looking at a screenful of text, and only part of the current paragraph you are reading is visible on the screen. The commands detailed in this section are used to shift which part of the current node is visible on the screen.

Scrolling commands are bound differently when '`--vi-keys`' operation is in effect (see [–vi-keys], page 4). These key bindings are designated with "vi-like operation".

⟨SPC⟩ (`scroll-forward`)
: Shift the text in this window up. That is, show more of the node which is currently below the bottom of the window. With a numeric argument, show that many more lines at the bottom of the window; a numeric argument of 4 would shift all of the text in the window up 4 lines (discarding the top 4 lines), and show you four new lines at the bottom of the window. Without a numeric argument, ⟨SPC⟩ takes the bottom two lines of the window and places them at the top of the window, redisplaying almost a completely new screenful of lines. If you are at the end of a node, ⟨SPC⟩ takes you to the "next" node, so that you can read an entire manual from start to finish by repeating ⟨SPC⟩.

  The default scroll size is one screen-full, but it can be changed by invoking the (`scroll-forward-page-only-set-window`) command, '`z`' under '`--vi-keys`', with a numeric argument.

⟨NEXT⟩ (an arrow key) (`scroll-forward-page-only`)
⟨C-v⟩
⟨C-f⟩, vi-like operation
⟨f⟩, vi-like operation
⟨M-SPC⟩, vi-like operation
: Shift the text in this window up. This is identical to the ⟨SPC⟩ operation above, except that it never scrolls beyond the end of the current node.

  The ⟨NEXT⟩ key is known as the ⟨PageDown⟩ key on some keyboards.

⟨z⟩ (`scroll-forward-page-only-set-window`, vi-like operation)
: Scroll forward, like with ⟨NEXT⟩, but if a numeric argument is specified, it becomes the default scroll size for subsequent `scroll-forward` and `scroll-backward` commands and their ilk.

⟨DEL⟩ (`scroll-backward`)
: Shift the text in this window down. The inverse of `scroll-forward`. If you are at the start of a node, ⟨DEL⟩ takes you to the "previous" node, so that you can read an entire manual from finish to start by repeating ⟨DEL⟩. The default scroll size can be changed by invoking the (`scroll-backward-page-only-set-window`) command, '`w`' under '`--vi-keys`', with a numeric argument.

⟨PREVIOUS⟩ (arrow key) (`scroll-backward-page-only`)
⟨PRIOR⟩ (arrow key)
⟨M-v⟩
⟨b⟩, vi-like operation
⟨C-b⟩, vi-like operation
: Shift the text in this window down. The inverse of `scroll-forward-page-only`. Does not scroll beyond the start of the current node. The default

scroll size can be changed by invoking the(`scroll-backward-page-only-set-window`) command, '`w`' under '`--vi-keys`', with a numeric argument.

⟨w⟩ (`scroll-backward-page-only-set-window`, vi-like operation)
: Scroll backward, like with ⟨PREVIOUS⟩, but if a numeric argument is specified, it becomes the default scroll size for subsequent `scroll-forward` and `scroll-backward` commands.

⟨C-n⟩ (`down-line`, vi-like operation)
⟨C-e⟩, vi-like operation
⟨RET⟩, vi-like operation
⟨LFD⟩, vi-like operation
⟨DOWN⟩, vi-like operation
: Scroll forward by one line. With a numeric argument, scroll forward that many lines.

⟨C-p⟩ (`up-line`, vi-like operation)
⟨UP⟩, vi-like operation
⟨y⟩, vi-like operation
⟨k⟩, vi-like operation
⟨C-k⟩, vi-like operation
⟨C-y⟩, vi-like operation
: Scroll backward one line. With a numeric argument, scroll backward that many lines.

⟨d⟩ (`scroll-half-screen-down`, vi-like operation)
⟨C-d⟩, vi-like operation
: Scroll forward by half of the screen size. With a numeric argument, scroll that many lines. If an argument is specified, it becomes the new default number of lines to scroll for subsequent '`d`' and '`u`' commands.

⟨u⟩ (`scroll-half-screen-up`, vi-like operation)
⟨C-u⟩, vi-like operation
: Scroll back by half of the screen size. With a numeric argument, scroll that many lines. If an argument is specified, it becomes the new default number of lines to scroll for subsequent '`u`' and '`d`' commands.

The `scroll-forward` and `scroll-backward` commands can also move forward and backward through the node structure of the file. If you press ⟨SPC⟩ while viewing the end of a node, or ⟨DEL⟩ while viewing the beginning of a node, what happens is controlled by the variable `scroll-behavior`. See Chapter 11 [Variables], page 26, for more information.

The `scroll-forward-page-only` and `scroll-backward-page-only` commands never scroll beyond the current node.

The ⟨PREVIOUS⟩ key is the ⟨PageUp⟩ key on many keyboards. Emacs refers to it by the name ⟨PRIOR⟩. When you use ⟨PRIOR⟩ or ⟨PageUp⟩ to scroll, Info never scrolls beyond the beginning of the current node.

If your keyboard lacks the ⟨DEL⟩ key, look for a key called ⟨BS⟩, or '`BackSpace`', sometimes designated with an arrow which points to the left, which should perform the same function.

⟨C-l⟩ (`redraw-display`)

Redraw the display from scratch, or shift the line containing the cursor to a specified location. With no numeric argument, '`C-l`' clears the screen, and then redraws its entire contents. Given a numeric argument of *n*, the line containing the cursor is shifted so that it is on the *n*th line of the window.

`C-x` ⟨w⟩ (`toggle-wrap`)

Toggles the state of line wrapping in the current window. Normally, lines which are longer than the screen width *wrap*, i.e., they are continued on the next line. Lines which wrap have a '\' appearing in the rightmost column of the screen. You can cause such lines to be terminated at the rightmost column by changing the state of line wrapping in the window with `C-x w`. When a line which needs more space than one screen width to display is displayed, a '`$`' appears in the rightmost column of the screen, and the remainder of the line is invisible. When long lines are truncated, the modeline displays the '`$`' character near its left edge.

5 Selecting a Node

This section details the numerous Info commands which select a new node to view in the current window.

The most basic node commands are '`n`', '`p`', '`u`', and '`l`'. Note that the commands to select nodes are mapped differently when '`--vi-keys`' is in effect; these keybindings are designated below as "vi-like operation".

When you are viewing a node, the top line of the node contains some Info *pointers* which describe where the next, previous, and up nodes are. Info uses this line to move about the node structure of the file when you use the following commands:

⟨n⟩ (`next-node`)
C-⟨NEXT⟩ (on DOS/Windows only)
C-*x* ⟨n⟩, vi-like operation
: Select the 'Next' node.

  The ⟨NEXT⟩ key is known as the ⟨PgDn⟩ key on some keyboards.

⟨p⟩ (`prev-node`)
C-⟨PREVIOUS⟩ (on DOS/Windows only)
: Select the 'Prev' node.

  The ⟨PREVIOUS⟩ key is known as the ⟨PgUp⟩ key on some keyboards.

⟨u⟩ (`up-node`)
C-⟨UP⟩ (an arrow key on DOS/Windows only)
C-*x* ⟨u⟩, vi-like operation
: Select the 'Up' node.

You can easily select a node that you have already viewed in this window by using the '`l`' command – this name stands for "last", and actually moves backwards through the history of visited nodes for this window. This is handy when you followed a reference to another node, possibly to read about a related issue, and would like then to resume reading at the same place where you started the excursion.

Each node where you press '`l`' is discarded from the history. Thus, by the time you get to the first node you visited in a window, the entire history of that window is discarded.

⟨l⟩ (`history-node`)
⟨C-⟨CENTER⟩⟩ (on DOS/Windows only)
⟨⟩, vi-like operation
: Pop the most recently selected node in this window from the node history.

Two additional commands make it easy to select the most commonly selected nodes; they are '`t`' and '`d`'.

⟨t⟩ (`top-node`)
⟨M-t⟩, vi-like operation
: Select the node '`Top`' in the current Info file.

⟨d⟩ (`dir-node`)
⟨M-d⟩, vi-like operation
: Select the directory node (i.e., the node '`(dir)`').

Here are some other commands which immediately result in the selection of a different node in the current window:

⟨<⟩ (`first-node`)
⟨g⟩, vi-like operation
: Selects the first node which appears in this file. This node is most often '`Top`', but it does not have to be. With a numeric argument *N*, select the *N*th node (the first node is node 1). An argument of zero is the same as the argument of 1.

⟨>⟩ (`last-node`)
⟨G⟩, vi-like operation
: Select the last node which appears in this file. With a numeric argument *N*, select the *N*th node (the first node is node 1). An argument of zero is the same as no argument, i.e., it selects the last node.

⟨]⟩ (`global-next-node`)
: Move forward or down through node structure. If the node that you are currently viewing has a '`Next`' pointer, that node is selected. Otherwise, if this node has a menu, the first menu item is selected. If there is no '`Next`' and no menu, the same process is tried with the '`Up`' node of this node.

⟨[⟩ (`global-prev-node`)
: Move backward or up through node structure. If the node that you are currently viewing has a '`Prev`' pointer, that node is selected. Otherwise, if the node has an '`Up`' pointer, that node is selected, and if it has a menu, the last item in the menu is selected.

You can get the same behavior as `global-next-node` and `global-prev-node` while simply scrolling through the file with ⟨SPC⟩ and ⟨DEL⟩; See Chapter 11 [Variables], page 26, for more information.

⟨g⟩ (`goto-node`)
C-x ⟨g⟩, vi-like operation
: Read the name of a node and select it. While reading the node name, completion (see Section 8.3 [The Echo Area], page 19) is only done for the nodes which reside in one of the Info files that were loaded in the current Info session; if the desired node resides in some other file, you must type the node exactly as it appears in that Info file, and you must include the Info file of the other file. For example,

```
g(emacs)Buffers
```

finds the node '`Buffers`' in the Info file '`emacs`'.

⟨O⟩ (`goto-invocation`
⟨I⟩
: Read the name of a program and look for a node in the current Info file which describes the invocation and the command-line options for that program. The default program name is derived from the name of the current Info file. This command does the same as the '`--show-options`' command-line option (see [–show-options], page 4), but it also allows to specify the program name; this is important for those manuals which describe several programs.

If you need to find the Invocation node of a program that is documented in another Info file, you need to visit that file before invoking '`I`'. For example, if you are reading the Emacs manual and want to see the command-line options of the `makeinfo` program, type *`g (texinfo)`* ⟨RET⟩ and then *`I makeinfo`* ⟨RET⟩. If you don't know what Info file documents the command, or if invoking '`I`' doesn't display the right node, go to the '`(dir)`' node (using the '`d`' command) and invoke '`I`' from there.

⟨G⟩ (`menu-sequence`)
: Read a sequence of menu entries and follow it. Info prompts for a sequence of menu items separated by commas. (Since commas are not allowed in a node name, they are a natural choice for a delimiter in a list of menu items.) Info then looks up the first item in the menu of the node '`(dir)`' (if the '`(dir)`' node cannot be found, Info uses '`Top`'). If such an entry is found, Info goes to the node it points to and looks up the second item in the menu of that node, etc. In other words, you can specify a complete path which descends through the menu hierarchy of a particular Info file starting at the '`(dir)`' node. This has the same effect as if you typed the menu item sequence on Info's command line, see [Info command-line arguments processing], page 4. For example,

  `G Texinfo,Overview,Reporting Bugs` ⟨RET⟩

  displays the node '`Reporting Bugs`' in the Texinfo manual. (You don't actually need to type the menu items in their full length, or in their exact letter-case. However, if you do type the menu items exactly, Info will find it faster.)

  If any of the menu items you type are not found, Info stops at the last entry it did find and reports an error.

`C-x` ⟨k⟩ (`kill-node`)
: Kill a node. The node name is prompted for in the echo area, with a default of the current node. *Killing* a node means that Info tries hard to forget about it, removing it from the list of history nodes kept for the window where that node is found. Another node is selected in the window which contained the killed node.

`C-x C-f` (`view-file`)
: Read the name of a file and selects the entire file. The command

  `C-x C-f` *`filename`*

  is equivalent to typing

  `g(filename)*`

`C-x C-b` (`list-visited-nodes`)
: Make a window containing a menu of all of the currently visited nodes. This window becomes the selected window, and you may use the standard Info commands within it.

`C-x` ⟨b⟩ (`select-visited-node`)
: Select a node which has been previously visited in a visible window. This is similar to '`C-x C-b`' followed by '`m`', but no window is created.

6 Searching an Info File

GNU Info allows you to search for a sequence of characters throughout an entire Info file, search through the indices of an Info file, or find areas within an Info file which discuss a particular topic.

⟨s⟩ (`search`)
⟨/⟩ Read a string in the echo area and search for it. If the string includes upper-case characters, the Info file is searched case-sensitively; otherwise Info ignores the letter case. With a numeric argument of *N*, search for *N*th occurrence of the string. Negative arguments search backwards.

⟨?⟩ (`search-backward`, vi-like operation)
Read a string in the echo area and search backward through the Info file for that string. If the string includes upper-case characters, the Info file is searched case-sensitively; otherwise Info ignores the letter case. With a numeric argument of *N*, search for *N*th occurrence of the string. Negative arguments search forward.

⟨S⟩ (`search-case-sensitively`
Read a string in the echo area and search for it case-sensitively, even if the string includes only lower-case letters. With a numeric argument of *N*, search for *N*th occurrence of the string. Negative arguments search backwards.

C-x ⟨n⟩ (`search-next`)
⟨n⟩, vi-like operation
Search for the same string used in the last search command, in the same direction, and with the same case-sensitivity option. With a numeric argument of *N*, search for *N*th next occurrence.

C-x ⟨N⟩ (`search-previous`)
⟨N⟩, vi-like operation
Search for the same string used in the last search command, and with the same case-sensitivity option, but in the reverse direction. With a numeric argument of *N*, search for *N*th previous occurrence.

⟨C-s⟩ (`isearch-forward`)
Interactively search forward through the Info file for a string as you type it. If the string includes upper-case characters, the search is case-sensitive; otherwise Info ignores the letter case.

⟨C-r⟩ (`isearch-backward`)
Interactively search backward through the Info file for a string as you type it. If the string includes upper-case characters, the search is case-sensitive; otherwise Info ignores the letter case.

⟨i⟩ (`index-search`)
Look up a string in the indices for this Info file, and select a node where the found index entry points to.

⟨,⟩ (`next-index-match`)
Move to the node containing the next matching index item from the last '`i`' command.

M-x index-apropos
: Grovel the indices of all the known Info files on your system for a string, and build a menu of the possible matches.

The most basic searching command is '`s`' or '`/`' (`search`). The '`s`' command prompts you for a string in the echo area, and then searches the remainder of the Info file for an occurrence of that string. If the string is found, the node containing it is selected, and the cursor is left positioned at the start of the found string. Subsequent '`s`' commands show you the default search string within '`[`' and '`]`'; pressing ⟨RET⟩ instead of typing a new string will use the default search string. Under '`--vi-keys`' (see [–vi-keys], page 4), using the '`n`' or '`N`' commands is a faster way of searching for the same string.

Incremental searching is similar to basic searching, but the string is looked up while you are typing it, instead of waiting until the entire search string has been specified.

Both incremental and non-incremental search by default ignore the case of letters when comparing the Info file text with the search string. However, an uppercase letter in the search string makes the search case-sensitive. You can force a case-sensitive non-incremental search, even for a string that includes only lower-case letters, by using the '`S`' command (`search-case-sensitively`). The '`n`' and '`N`' commands operate case-sensitively if the last search command was '`S`'.

The most efficient means of finding something quickly in a manual is the '`i`' command (`index-search`). This command prompts for a string, and then looks for that string in all the indices of the current Info manual. If it finds a matching index entry, it displays the node to which that entry refers and prints the full text of the entry in the echo area. You can press '`,`' (`next-index-match`) to find more matches. A good Info manual has all of its important concepts indexed, so the '`i`' command lets you use a manual as a reference.

If you don't know what manual documents something, try the *M-x index-apropos*. It prompts for a string and then looks up that string in all the indices of all the Info documents installed on your system. It can also be invoked from the command line; see [–apropos], page 2.

7 Selecting Cross References

We have already discussed the '`Next`', '`Prev`', and '`Up`' pointers which appear at the top of a node. In addition to these pointers, a node may contain other pointers which refer you to a different node, perhaps in another Info file. Such pointers are called *cross references*, or *xrefs* for short.

7.1 Parts of an Xref

Cross references have two major parts: the first part is called the *label*; it is the name that you can use to refer to the cross reference, and the second is the *target*; it is the full name of the node that the cross reference points to.

The target is separated from the label by a colon '`:`'; first the label appears, and then the target. For example, in the sample menu cross reference below, the single colon separates the label from the target.

```
* Foo Label: Foo Target.        More information about Foo.
```

Note the '`.`' which ends the name of the target. The '`.`' is not part of the target; it serves only to let Info know where the target name ends.

A shorthand way of specifying references allows two adjacent colons to stand for a target name which is the same as the label name:

```
* Foo Commands::                Commands pertaining to Foo.
```

In the above example, the name of the target is the same as the name of the label, in this case `Foo Commands`.

You will normally see two types of cross reference while viewing nodes: *menu* references, and *note* references. Menu references appear within a node's menu; they begin with a '`*`' at the beginning of a line, and continue with a label, a target, and a comment which describes what the contents of the node pointed to contains.

Note references appear within the body of the node text; they begin with `*Note`, and continue with a label and a target.

Like '`Next`', '`Prev`', and '`Up`' pointers, cross references can point to any valid node. They are used to refer you to a place where more detailed information can be found on a particular subject. Here is a cross reference which points to a node within the Texinfo documentation: See section "Writing an Xref" in *the Texinfo Manual*, for more information on creating your own texinfo cross references.

7.2 Selecting Xrefs

The following table lists the Info commands which operate on menu items.

⟨1⟩ (`menu-digit`)
⟨2⟩ ... ⟨9⟩
⟨M-1⟩, vi-like operation
⟨M-2⟩ ... ⟨M-9⟩, vi-like operation

Within an Info window, pressing a single digit, (such as '`1`'), selects that menu item, and places its node in the current window. For convenience, there is

one exception; pressing '0' selects the *last* item in the node's menu. When '--vi-keys' is in effect, digits set the numeric argument, so these commands are remapped to their 'M-' varieties. For example, to select the last menu item, press ⟨M-0⟩.

⟨0⟩ (last-menu-item)
⟨M-0⟩, vi-like operation
Select the last item in the current node's menu.

⟨m⟩ (menu-item)
Reads the name of a menu item in the echo area and selects its node. Completion is available while reading the menu label. See Section 8.3 [The Echo Area], page 19.

M-x find-menu
Move the cursor to the start of this node's menu.

This table lists the Info commands which operate on cross references.

⟨f⟩ (xref-item)
⟨r⟩

⟨M-f⟩, vi-like operation
C-x ⟨r⟩, vi-like operation
Reads the name of a note cross reference in the echo area and selects its node. Completion is available while reading the cross reference label. See Section 8.3 [The Echo Area], page 19.

Finally, the next few commands operate on menu or note references alike:

⟨TAB⟩ (move-to-next-xref)
Move the cursor to the start of the next nearest menu item or note reference in this node. You can then use ⟨RET⟩ (select-reference-this-line) to select the menu or note reference.

⟨M-TAB⟩ (move-to-prev-xref)
⟨Shift-⟨TAB⟩⟩ (on DOS/Windows only)
Move the cursor the start of the nearest previous menu item or note reference in this node.

On DOS/Windows only, the *Shift-*⟨TAB⟩ key is an alias for *M-*⟨TAB⟩. This key is sometimes called 'BackTab'.

⟨RET⟩ (select-reference-this-line)
⟨M-g⟩, vi-like operation
Select the menu item or note reference appearing on this line.

8 Manipulating Multiple Windows

A *window* is a place to show the text of a node. Windows have a view area where the text of the node is displayed, and an associated *mode line*, which briefly describes the node being viewed.

GNU Info supports multiple windows appearing in a single screen; each window is separated from the next by its modeline. At any time, there is only one *active* window, that is, the window in which the cursor appears. There are commands available for creating windows, changing the size of windows, selecting which window is active, and for deleting windows.

8.1 The Mode Line

A *mode line* is a line of inverse video which appears at the bottom of an Info window. It describes the contents of the window just above it; this information includes the name of the file and node appearing in that window, the number of screen lines it takes to display the node, and the percentage of text that is above the top of the window. It can also tell you if the indirect tags table for this Info file needs to be updated, and whether or not the Info file was compressed when stored on disk.

Here is a sample mode line for a window containing an uncompressed file named '`dir`', showing the node '`Top`'.

```
-----Info: (dir)Top, 40 lines --Top-----------------------------------
            ^^   ^   ^^^         ^^

          (file)Node #lines    where
```

When a node comes from a file which is compressed on disk, this is indicated in the mode line with two small '`z`'s. In addition, if the Info file containing the node has been split into subfiles, the name of the subfile containing the node appears in the modeline as well:

```
--zz-Info: (emacs)Top, 291 lines --Top-- Subfile: emacs-1.Z-------------
```

Truncation of long lines (as opposed to wrapping them to the next display line, see Chapter 4 [Scrolling Commands], page 8) is indicated by a '`$`' at the left edge of the mode line:

```
--$--Info: (texinfo)Top, 480 lines --Top-- Subfile: texinfo-1-----------
```

When Info makes a node internally, such that there is no corresponding info file on disk, the name of the node is surrounded by asterisks ('`*`'). The name itself tells you what the contents of the window are; the sample mode line below shows an internally constructed node showing possible completions:

```
-----Info: *Completions*, 7 lines --All--------------------------------
```

8.2 Window Commands

It can be convenient to view more than one node at a time. To allow this, Info can display more than one *window*. Each window has its own mode line (see Section 8.1 [The Mode Line], page 18) and history of nodes viewed in that window (see Chapter 5 [`history-node`], page 11).

C-x ⟨o⟩ (`next-window`)
: Select the next window on the screen. Note that the echo area can only be selected if it is already in use, and you have left it temporarily. Normally, '`C-x o`' simply moves the cursor into the next window on the screen, or if you are already within the last window, into the first window on the screen. Given a numeric argument, '`C-x o`' moves over that many windows. A negative argument causes '`C-x o`' to select the previous window on the screen.

M-x prev-window
: Select the previous window on the screen. This is identical to '`C-x o`' with a negative argument.

C-x ⟨2⟩ (`split-window`)
: Split the current window into two windows, both showing the same node. Each window is one half the size of the original window, and the cursor remains in the original window. The variable `automatic-tiling` can cause all of the windows on the screen to be resized for you automatically, please see Chapter 11 [automatic-tiling], page 26 for more information.

C-x ⟨0⟩ (`delete-window`)
: Delete the current window from the screen. If you have made too many windows and your screen appears cluttered, this is the way to get rid of some of them.

C-x ⟨1⟩ (`keep-one-window`)
: Delete all of the windows excepting the current one.

ESC ⟨C-v⟩ (`scroll-other-window`)
: Scroll the other window, in the same fashion that '`C-v`' might scroll the current window. Given a negative argument, scroll the `"other"` window backward.

C-x ⟨^⟩ (`grow-window`)
: Grow (or shrink) the current window. Given a numeric argument, grow the current window that many lines; with a negative numeric argument, shrink the window instead.

C-x ⟨t⟩ (`tile-windows`)
: Divide the available screen space among all of the visible windows. Each window is given an equal portion of the screen in which to display its contents. The variable `automatic-tiling` can cause `tile-windows` to be called when a window is created or deleted. See Chapter 11 [`automatic-tiling`], page 26.

8.3 The Echo Area

The *echo area* is a one line window which appears at the bottom of the screen. It is used to display informative or error messages, and to read lines of input from you when that is necessary. Almost all of the commands available in the echo area are identical to their Emacs counterparts, so please refer to that documentation for greater depth of discussion on the concepts of editing a line of text. The following table briefly lists the commands that are available while input is being read in the echo area:

⟨C-f⟩ (`echo-area-forward`)
⟨RIGHT⟩ (an arrow key)
⟨M-h⟩, vi-like operation
: Move forward a character.

⟨C-b⟩ (`echo-area-backward`)
⟨LEFT⟩ (an arrow key)
⟨M-l⟩, vi-like operation
: Move backward a character.

⟨C-a⟩ (`echo-area-beg-of-line`)
⟨M-0⟩, vi-like operation
: Move to the start of the input line.

⟨C-e⟩ (`echo-area-end-of-line`)
⟨M-$⟩, vi-like operation
: Move to the end of the input line.

⟨M-f⟩ (`echo-area-forward-word`)
⟨C-⟨RIGHT⟩⟩ (DOS/Windows only)
⟨M-w⟩, vi-like operation
: Move forward a word.

  On DOS/Windows, `C-`⟨RIGHT⟩ moves forward by words.

⟨M-b⟩ (`echo-area-backward-word`)
⟨C-⟨LEFT⟩⟩ (DOS/Windows only)
: Move backward a word.

  On DOS/Windows, `C-`⟨LEFT⟩ moves backward by words.

⟨C-d⟩ (`echo-area-delete`)
⟨M-x⟩, vi-like operation
: Delete the character under the cursor.

⟨DEL⟩ (`echo-area-rubout`)
: Delete the character behind the cursor.

  On some keyboards, this key is designated ⟨BS⟩, for '`BackSpace`'. Those keyboards will usually bind ⟨DEL⟩ in the echo area to `echo-area-delete`.

⟨C-g⟩ (`echo-area-abort`)
⟨C-u⟩, vi-like operation
: Cancel or quit the current operation. If completion is being read, this command discards the text of the input line which does not match any completion. If the input line is empty, it aborts the calling function.

⟨RET⟩ (`echo-area-newline`)
: Accept (or forces completion of) the current input line.

⟨C-q⟩ (`echo-area-quoted-insert`)
⟨C-v⟩, vi-like operation
: Insert the next character verbatim. This is how you can insert control characters into a search string, for example, or the '`?`' character when Info prompts with completion.

printing character (`echo-area-insert`)
: Insert the character. Characters that have their 8th bit set, and not bound to '`M-`' commands, are also inserted verbatim; this is useful for terminals which support Latin scripts.

⟨M-TAB⟩ (`echo-area-tab-insert`)
⟨Shift-⟨TAB⟩⟩ (on DOS/Windows only)
: Insert a TAB character.

  On DOS/Windows only, the *`Shift`*`-`⟨TAB⟩ key is an alias for *`M`*`-`⟨TAB⟩. This key is sometimes called '`BackTab`'.

⟨C-t⟩ (`echo-area-transpose-chars`)
: Transpose the characters at the cursor.

The next group of commands deal with *killing*, and *yanking* text[1]. For an in-depth discussion of killing and yanking, see section "Killing and Deleting" in *the GNU Emacs Manual*.

⟨M-d⟩ (`echo-area-kill-word`)
⟨M-X⟩, vi-like operation
: Kill the word following the cursor.

⟨M-DEL⟩ (`echo-area-backward-kill-word`)
⟨M-⟨BS⟩⟩
: Kill the word preceding the cursor.

  On some keyboards, the `Backspace` key is used instead of `DEL`, so `M-`⟨Backspace⟩ has the same effect as `M-`⟨DEL⟩.

⟨C-k⟩ (`echo-area-kill-line`)
: Kill the text from the cursor to the end of the line.

`C-x` ⟨DEL⟩ (`echo-area-backward-kill-line`)
: Kill the text from the cursor to the beginning of the line.

⟨C-y⟩ (`echo-area-yank`)
: Yank back the contents of the last kill.

⟨M-y⟩ (`echo-area-yank-pop`)
: Yank back a previous kill, removing the last yanked text first.

Sometimes when reading input in the echo area, the command that needed input will only accept one of a list of several choices. The choices represent the *possible completions*, and you must respond with one of them. Since there are a limited number of responses you can make, Info allows you to abbreviate what you type, only typing as much of the response as is necessary to uniquely identify it. In addition, you can request Info to fill in as much of the response as is possible; this is called *completion*.

The following commands are available when completing in the echo area:

⟨TAB⟩ (`echo-area-complete`)
⟨SPC⟩
: Insert as much of a completion as is possible.

[1] Some people are used to calling these operations *cut* and *paste*, respectively.

? (`echo-area-possible-completions`)
: Display a window containing a list of the possible completions of what you have typed so far. For example, if the available choices are:

```
bar
foliate
food
forget
```

and you have typed an '`f`', followed by '`?`', Info will pop up a window showing a node called '`*Completions*`' which lists the possible completions like this:

```
3 completions:
foliate         food
forget
```

i.e., all of the choices which begin with '`f`'. Pressing ⟨SPC⟩ or ⟨TAB⟩ would result in '`fo`' appearing in the echo area, since all of the choices which begin with '`f`' continue with '`o`'. Now, typing '`l`' followed by '`TAB`' results in '`foliate`' appearing in the echo area, since that is the only choice which begins with '`fol`'.

⟨ESC C-v⟩ (`echo-area-scroll-completions-window`)
: Scroll the completions window, if that is visible, or the "other" window if not.

9 Printing Nodes

In general, we recommend that you use TEX to format the document and print sections of it, by running `tex` on the Texinfo source file. However, you may wish to print out the contents of a node as a quick reference document for later use, or if you don't have TEX installed. Info provides you with a command for doing this.

M-x print-node

Pipe the contents of the current node through the command in the environment variable `INFO_PRINT_COMMAND`. If the variable does not exist, the node is simply piped to `lpr` (on DOS/Windows, the default is to print the node to the local printer device, '`PRN`').

The value of `INFO_PRINT_COMMAND` may begin with the '`>`' character, as in '`>/dev/printer`', in which case Info treats the rest as the name of a file or a device. Instead of piping to a command, Info opens the file, writes the node contents, and closes the file, under the assumption that text written to that file will be printed by the underlying OS.

10 Miscellaneous Commands

GNU Info contains several commands which self-document GNU Info:

M-x describe-command
: Read the name of an Info command in the echo area and then display a brief description of what that command does.

M-x describe-key
: Read a key sequence in the echo area, and then display the name and documentation of the Info command that the key sequence invokes.

M-x describe-variable
: Read the name of a variable in the echo area and then display a brief description of what the variable affects.

M-x where-is
: Read the name of an Info command in the echo area, and then display a key sequence which can be typed in order to invoke that command.

⟨C-h⟩ (`get-help-window`)
⟨?⟩
⟨F1⟩ (on DOS/Windows only)
h, vi-like operation
: Create (or Move into) the window displaying `*Help*`, and place a node containing a quick reference card into it. This window displays the most concise information about GNU Info available.

⟨h⟩ (`get-info-help-node`)
⟨M-h⟩, vi-like operation
: Try hard to visit the node `(info)Help`. The Info file '`info.texi`' distributed with GNU Info contains this node. Of course, the file must first be processed with `makeinfo`, and then placed into the location of your Info directory.

Here are the commands for creating a numeric argument:

⟨C-u⟩ (`universal-argument`)
: Start (or multiply by 4) the current numeric argument. '`C-u`' is a good way to give a small numeric argument to cursor movement or scrolling commands; '`C-u C-v`' scrolls the screen 4 lines, while '`C-u C-u C-n`' moves the cursor down 16 lines. '`C-u`' followed by digit keys sets the numeric argument to the number thus typed: *`C-u 1 2 0`* sets the argument to 120.

⟨M-1⟩ (`add-digit-to-numeric-arg`)
⟨1⟩, vi-like operation
⟨M-2⟩ ... ⟨M-9⟩
⟨2⟩ ... ⟨9⟩, vi-like operation
⟨M-0⟩
⟨0⟩, vi-like operation
: Add the digit value of the invoking key to the current numeric argument. Once Info is reading a numeric argument, you may just type the digits of the argument, without the Meta prefix. For example, you might give '`C-l`' a numeric argument of 32 by typing:

C-u 3 2 C-l

or

M-3 2 C-l

⟨M–⟩ (`add-digit-to-numeric-arg`
⟨-⟩ To make a negative argument, type `-`. Typing `-` alone makes a negative argument with a value of -1. If you continue to type digit or Meta-digit keys after `-`, the result is a negative number produced by those digits.

`-` doesn't work when you type in the echo area, because you need to be able to insert the '`-`' character itself; use *M--* instead, if you need to specify negative arguments in the echo area.

'`C-g`' is used to abort the reading of a multi-character key sequence, to cancel lengthy operations (such as multi-file searches) and to cancel reading input in the echo area.

⟨C-g⟩ (`abort-key`)
⟨C-u⟩, vi-like operation
Cancel current operation.

The '`q`' command of Info simply quits running Info. Under '`--vi-keys`' (see [–vi-keys], page 4), you can also exit with '`:q`' or '`ZZ`'.

⟨q⟩ (`quit`)
C-x C-c
:q, vi-like operation
ZZ, vi-like operation
Exit GNU Info.

If the operating system tells GNU Info that the screen is 60 lines tall, and it is actually only 40 lines tall, here is a way to tell Info that the operating system is correct.

M-x set-screen-height
Read a height value in the echo area and set the height of the displayed screen to that value.

On MS-DOS/MS-Windows, this command actually tries to change the dimensions of the visible screen to the value you type in the echo area.

Finally, Info provides a convenient way to display footnotes which might be associated with the current node that you are viewing:

⟨ESC C-f⟩ (`show-footnotes`)
Show the footnotes (if any) associated with the current node in another window. You can have Info automatically display the footnotes associated with a node when the node is selected by setting the variable `automatic-footnotes`. See Chapter 11 [`automatic-footnotes`], page 26.

11 Manipulating Variables

GNU Info contains several *variables* whose values are looked at by various Info commands. You can change the values of these variables, and thus change the behavior of Info to more closely match your environment and Info file reading manner.

There are two ways to set the value of a variable: interactively, using the `set-variable` command described below, or in the `#var` section of the `.infokey` file. See Chapter 12 [Custom Key Bindings], page 29.

M-x set-variable
: Read the name of a variable, and the value for it, in the echo area and then set the variable to that value. Completion is available when reading the variable name (see Section 8.3 [The Echo Area], page 19); often, completion is available when reading the value to give to the variable, but that depends on the variable itself. If a variable does *not* supply multiple choices to complete over, it expects a numeric value.

M-x describe-variable
: Read the name of a variable in the echo area and then display a brief description of what the variable affects.

Here is a list of the variables that you can set in Info.

`automatic-footnotes`
: When set to `On`, footnotes appear and disappear automatically; else, they appear at the bottom of the node text. This variable is `Off` by default. When a node is selected, a window containing the footnotes which appear in that node is created, and the footnotes are displayed within the new window. The window that Info creates to contain the footnotes is called '`*Footnotes*`'. If a node is selected which contains no footnotes, and a '`*Footnotes*`' window is on the screen, the '`*Footnotes*`' window is deleted. Footnote windows created in this fashion are not automatically tiled so that they can use as little of the display as is possible.

`automatic-tiling`
: When set to `On`, creating or deleting a window resizes other windows. This variable is `Off` by default. Normally, typing '`C-x 2`' divides the current window into two equal parts. When `automatic-tiling` is set to `On`, all of the windows are resized automatically, keeping an equal number of lines visible in each window. There are exceptions to the automatic tiling; specifically, the windows '`*Completions*`' and '`*Footnotes*`' are *not* resized through automatic tiling; they remain their original size.

`errors-ring-bell`
: When set to `On`, errors cause the bell to ring. The default setting of this variable is `On`.

`gc-compressed-files`
: When set to `On`, Info garbage collects files which had to be uncompressed. The default value of this variable is `Off`. Whenever a node is visited in Info, the Info

file containing that node is read into core, and Info reads information about the tags and nodes contained in that file. Once the tags information is read by Info, it is never forgotten. However, the actual text of the nodes does not need to remain in core unless a particular Info window needs it. For non-compressed files, the text of the nodes does not remain in core when it is no longer in use. But de-compressing a file can be a time consuming operation, and so Info tries hard not to do it twice. `gc-compressed-files` tells Info it is okay to garbage collect the text of the nodes of a file which was compressed on disk.

`ISO-Latin`
When set to `On`, Info accepts and displays ISO Latin characters. By default, Info assumes an ASCII character set. `ISO-Latin` tells Info that it is running in an environment where the European standard character set is in use, and allows you to input such characters to Info, as well as display them.

`scroll-behavior`
Control what happens when forward scrolling is requested at the end of a node, or when backward scrolling is requested at the beginning of a node. The default value for this variable is `Continuous`. There are three possible values for this variable:

`Continuous`
Try to get the first item in this node's menu, or failing that, the '`Next`' node, or failing that, the '`Next`' of the '`Up`'. This behavior is identical to using the '`]`' (`global-next-node`) and '`[`' (`global-prev-node`) commands.

`Next Only` Only try to get the '`Next`' node.

`Page Only` Simply give up, changing nothing. If `scroll-behavior` is `Page Only`, no scrolling command can change the node that is being viewed.

`scroll-step`
The number of lines to scroll when the cursor moves out of the window. Scrolling happens automatically if the cursor has moved out of the visible portion of the node text when it is time to display. Usually the scrolling is done so as to put the cursor on the center line of the current window. However, if the variable `scroll-step` has a nonzero value, Info attempts to scroll the node text by that many lines; if that is enough to bring the cursor back into the window, that is what is done. The default value of this variable is 0, thus placing the cursor (and the text it is attached to) in the center of the window. Setting this variable to 1 causes a kind of "smooth scrolling" which some people prefer.

`show-index-match`
When set to `On`, the portion of the matched search string is highlighted in the message which explains where the matched search string was found. The default value of this variable is `On`. When Info displays the location where an index match was found, (see Chapter 6 [`next-index-match`], page 14), the portion of the string that you had typed is highlighted by displaying it in the inverse case from its surrounding characters.

`visible-bell`

When set to `On`, GNU Info attempts to flash the screen instead of ringing the bell. This variable is `Off` by default. Of course, Info can only flash the screen if the terminal allows it; in the case that the terminal does not allow it, the setting of this variable has no effect. However, you can make Info perform quietly by setting the `errors-ring-bell` variable to `Off`.

12 Customizing Key Bindings and Variables

For those whose editor/pager of choice is not Emacs and who are not entirely satisfied with the –vi-keys option (see [–vi-keys], page 4), GNU Info provides a way to define different key-to-command bindings and variable settings from the defaults described in this document.

On startup, GNU Info looks for a configuration file in the invoker's HOME directory called '`.info`'[1]. If it is present, and appears to contain Info configuration data, and was created with the current version of the `infokey` command, then Info adopts the key bindings and variable settings contained therein.

The '`.info`' file contains compact, non-textual data for reasons of efficiency and because its design was lifted wholesale from the GNU Less program, which also does it that way. It must be created by compiling a textual source file using the `infokey` command.

12.1 Invoking `infokey`

`infokey` compiles a source file ('`$HOME/.infokey`'[2] by default) containing Info customizations into a binary format ('`$HOME/.info`' by default). GNU Info reads the binary file at startup to override the default key bindings and variable definitions. Synopsis:

```
infokey [option...] [input-file]
```

Besides the standard '`--help`' and '`--version`', the only option is '`--output` *file*'. This tells `infokey` to write the binary data to *file* instead of '`$HOME/.info`'.

12.2 `infokey` source format

The format of the source file read by `infokey` is most easily illustrated by example. For instance, here is a sample '`.infokey`' source file suitable for aficionados of `vi` or `less`:

```
#info
j       next-line
k       prev-line
l       forward-char
h       backward-char
\kd     next-line
\ku     prev-line
\kr     forward-char
\kl     backward-char
\       scroll-forward
\kD     scroll-forward-page-only
b       scroll-backward
\kU     scroll-backward-page-only
```

[1] Due to the limitations of DOS filesystems, the MS-DOS version of Info looks for a file '`_info`' instead. If the `HOME` variable is not defined, Info additionally looks in the current directory.

[2] This file is named '`_infokey`' in the MS-DOS version, and is looked for in the current directory if `HOME` is undefined.

```
g       beginning-of-node
\kh     beginning-of-node
G       end-of-node
\ke     end-of-node
\t      select-reference-this-line
-       history-node
n       next-node
p       prev-node
u       up-node
t       top-node
d       dir-node
#var
scroll-step=1
```

The source file consists of one or more *sections*. Each section starts with a line that identifies the type of section. Possible sections are:

`#info` Key bindings for Info windows. The start of this section is indicated by a line containing just `#info` by itself. If this is the first section in the source file, the `#info` line can be omitted. The rest of this section consists of lines of the form:

string `whitespace` *action* `[ whitespace [ # comment ] ] newline`

Whitespace is any sequence of one or more spaces and/or tabs. Comment is any sequence of any characters, excluding newline. *string* is the key sequence which invokes the action. *action* is the name of an Info command. The characters in *string* are interpreted literally or prefixed by a caret (`^`) to indicate a control character. A backslash followed by certain characters specifies input keystrokes as follows:

`\b`	Backspace
`\e`	Escape (ESC)
`\n`	Newline
`\r`	Return
`\t`	Tab
`\ku`	Up arrow
`\kd`	Down arrow
`\kl`	Left arrow
`\kr`	Right arrow
`\kU`	Page Up
`\kD`	Page Down
`\kh`	HOME
`\ke`	END
`\kx`	Delete (DEL)

`\mx` Meta-*x* where *x* is any character as described above.

Backslash followed by any other character indicates that character is to be taken literally. Characters which must be preceded by a backslash include caret, space, tab, and backslash itself.

`#echo-area`
Key bindings for the echo area. The start of this section is indicated by a line containing just `#echo-area` by itself. The rest of this section has a syntax identical to that for the key definitions for the Info area, described above.

`#var` Variable initializations. The start of this section is indicated by a line containing just `#var` by itself. Following this line is a list of variable assignments, one per line. Each line consists of a variable name (See Chapter 11 [Variables], page 26,) followed by `=` followed by a value. There may be no white space between the variable name and the `=`, and all characters following the `=`, including white space, are included in the value.

Blank lines and lines starting with `#` are ignored, except for the special section header lines.

Key bindings defined in the '`.info`' file take precedence over GNU Info's default key bindings, whether or not '`--vi-keys`' is used. A default key binding may be disabled by overriding it in the '`.info`' file with the action `invalid`. In addition, *all* default key bindings can be disabled by adding this line *anywhere* in the relevant section:

```
#stop
```

This will cause GNU Info to ignore all the default key commands for that section.

Beware: `#stop` can be dangerous. Since it disables all default key bindings, you must supply enough new key bindings to enable all necessary actions. Failure to bind any key to the `quit` command, for example, can lead to frustration.

The order in which key bindings are defined in the '`.info`' file is not important, except that the command summary produced by the `get-help-window` command only displays the *first* key that is bound to each command.

Appendix A Copying This Manual

A.1 GNU Free Documentation License

Version 1.2, November 2002

Copyright © 2000,2001,2002 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related

matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

   You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

   You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

   If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

   If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

   If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

   It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

   You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
H. Include an unaltered copy of this License.
I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at

your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

   You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

   The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

   In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warrany Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.1.1 ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled
``GNU Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Appendix B Index

'

'`, vi-like operation` 11

,

`,` 14

-

`-` 25
`--subnodes`, command line option 4

.

.info 29
.infokey 29
.infokey source format 29

/

`/` 14

<

`<` 12

>

`>` 12

?

`?, in Info windows` 24
`?, in the echo area` 22
`?, vi-like operation` 14

[

`[` 12

]

`]` 12

_

_info file (MS-DOS) 29
_infokey file (MS-DOS) 29

0

`0 ... 9, vi-like operation` 24
`0, in Info windows` 17

1

`1 ... 9, in Info windows` 16

A

`abort-key` 25
`add-digit-to-numeric-arg` 24
ANSI escape sequences in man pages 3
Apropos, in Info files 2
arguments, command line 2
arguments, negative 25
`automatic-footnotes` 26
`automatic-tiling` 26

B

`b, in Info windows` 7
`b, vi-like operation` 8
`BackTab, in Info windows` 17
`BackTab, in the echo area` 21
`backward-char` 6
`backward-word` 7
`beginning-of-line` 6
`beginning-of-node` 7
`BS (backspace)` 9

C

`C-a, in Info windows` 6
`C-a, in the echo area` 20
`C-b, in Info windows` 6
`C-b, in the echo area` 20
`C-b, vi-like operation` 8
`C-CENTER` 11
`C-d, in the echo area` 20
`C-d, vi-like operation` 9
`C-e, in Info windows` 6
`C-e, in the echo area` 20
`C-e, vi-like operation` 9
`C-End` 7
`C-f, in Info windows` 6
`C-f, in the echo area` 20
`C-f, vi-like operation` 8
`C-g, in Info windows` 25
`C-g, in the echo area` 20
`C-h` 24
`C-Home` 7
`C-k, in the echo area` 21
`C-k, vi-like operation` 9
`C-l` 10
`C-LEFT` 7
`C-LEFT, in the echo area` 20
`C-n` 6
`C-n, vi-like operation` 9

C-NEXT ... 11
C-p ... 6
C-p, vi-like operation ... 9
C-PgDn ... 11
C-PgUp ... 11
C-PREVIOUS ... 11
C-q, in the echo area ... 20
C-r ... 14
C-RIGHT ... 6
C-RIGHT, in the echo area ... 20
C-s ... 14
C-t, in the echo area ... 21
C-u ... 24
C-u cancels typeahead, vi-like operation ... 25
C-u, in the echo area, vi-like operation ... 20
C-u, vi-like operation ... 9
C-UP ... 11
C-v ... 8
C-v, in the echo area, vi-like operation ... 20
C-w ... 10
C-x ^ ... 19
C-x 0 ... 19
C-x 1 ... 19
C-x 2 ... 19
C-x b ... 13
C-x C-b ... 13
C-x C-c ... 25
C-x C-f ... 13
C-x DEL, in the echo area ... 21
C-x g, vi-like operation ... 12
C-x k ... 13
C-x n ... 14
C-x N ... 14
C-x n, vi-like operation ... 11
C-x o ... 19
C-x r, vi-like operation ... 17
C-x t ... 19
C-x u, vi-like operation ... 11
C-y, in the echo area ... 21
C-y, vi-like operation ... 9
cancelling the current operation ... 25
cancelling typeahead ... 25
case-sensitive search ... 14
case-sensitivity, and search ... 15
colors in man pages ... 3
command line options ... 2
command-line options, how to find ... 4
commands, describing ... 24
completion ... 21
compressed Info files ... 2
cursor, moving ... 6
customizing key bindings ... 29

D

d ... 11
d, vi-like operation ... 9
default key bindings, overriding ... 29
DEL, in Info windows ... 8
DEL, in the echo area ... 20
delete-window ... 19
describe-command ... 24
describe-key ... 24
describe-variable ... 26
dir-node ... 11
directory path ... 2
DOWN (an arrow key) ... 6
DOWN, vi-like operation ... 9
down-line ... 9

E

e, in Info windows ... 7
echo area ... 19
echo-area-abort ... 20
echo-area-backward ... 20
echo-area-backward-kill-line ... 21
echo-area-backward-kill-word ... 21
echo-area-backward-word ... 20
echo-area-beg-of-line ... 20
echo-area-complete ... 21
echo-area-delete ... 20
echo-area-end-of-line ... 20
echo-area-forward ... 20
echo-area-forward-word ... 20
echo-area-insert ... 21
echo-area-kill-line ... 21
echo-area-kill-word ... 21
echo-area-newline ... 20
echo-area-possible-completions ... 22
echo-area-quoted-insert ... 20
echo-area-rubout ... 20
echo-area-scroll-completions-window ... 22
echo-area-tab-insert ... 21
echo-area-transpose-chars ... 21
echo-area-yank ... 21
echo-area-yank-pop ... 21
End ... 6
end-of-line ... 6
end-of-node ... 7
errors-ring-bell ... 26
ESC C-f ... 25
ESC C-v, in Info windows ... 19
ESC C-v, in the echo area ... 22

F

f ... 17
f, vi-like operation ... 8
F1 ... 24
FDL, GNU Free Documentation License ... 32
file names, relative ... 2

file, outputting to ... 3
files, compressed ... 2
find-menu ... 17
finding the Invocation node ... 12
first-node ... 12
footnotes, displaying ... 25
format of .infokey source ... 29
forward-char ... 6
forward-word ... 6
functions, describing ... 24

G

g ... 12
G ... 13
g, vi-like operation ... 12
G, vi-like operation ... 12
gc-compressed-files ... 26
get-help-window ... 24
get-info-help-node ... 24
global-next-node ... 12
global-prev-node ... 12
goto-invocation ... 12
goto-node ... 12
grow-window ... 19

H

h ... 24
h, vi-like operation ... 24
history-node ... 11
Home ... 6

I

i ... 14
I ... 12
incremental search ... 14
index search, selecting from the command line ... 3
index, searching ... 14
index-apropos ... 15
index-search ... 14
Info file, selecting ... 2
Info files, compressed ... 2
Info files, relative ... 2
Info files, searching all indices ... 2
Info, invoking ... 2
INFO_PRINT_COMMAND, environment variable ... 23
infokey ... 29
infokey source format ... 29
infokey, invoking ... 29
invocation description, how to find ... 4
invoking Info ... 2
invoking infokey ... 29
isearch-backward ... 14
isearch-forward ... 14
ISO Latin characters ... 27
ISO-Latin ... 27

K

k, vi-like operation ... 9
keep-one-window ... 19
key bindings, customizing ... 29
keys, describing ... 24
keystrokes, recording ... 2
kill-node ... 13

L

l ... 11
last-menu-item ... 17
last-node ... 12
LEFT (an arrow key) ... 6
LEFT, in the echo area ... 20
Less-like key bindings ... 4
LFD, vi-like operation ... 9
list-visited-nodes ... 13
local printer device ... 23

M

m ... 17
M-$, vi-like operation ... 20
M-- ... 25
M-< ... 7
M-> ... 7
M-0 ... M-9 ... 24
M-0, in the echo area, vi-like operation ... 20
M-0, vi-like operation ... 17
M-1 ... M-9, vi-like operation ... 16
M-b, in Info windows ... 7
M-b, in the echo area ... 20
M-b, vi-like operation ... 7
M-BS, in the echo area ... 21
M-d, in the echo area ... 21
M-d, vi-like operation ... 11
M-DEL, in the echo area ... 21
M-f, in Info windows ... 6
M-f, in the echo area ... 20
M-f, vi-like operation ... 17
M-g, vi-like operation ... 17
M-h, in the echo area, vi-like operation ... 20
M-h, vi-like operation ... 24
M-l, in the echo area, vi-like operation ... 20
M-r ... 7
M-SPC, vi-like operation ... 8
M-t, vi-like operation ... 11
M-TAB, in Info windows ... 17
M-TAB, in the echo area ... 21
M-v ... 8
M-w, in the echo area, vi-like operation ... 20
M-x, in the echo area, vi-like operation ... 20
M-X, in the echo area, vi-like operation ... 21
M-y, in the echo area ... 21

menu, following ... 4
menu, following, from inside Info ... 13
menu-digit ... 16
menu-item ... 17
menu-sequence ... 13
move-to-next-xref ... 17
move-to-prev-xref ... 17
move-to-window-line ... 7
moving the cursor ... 6

N

n ... 11
n, vi-like operation ... 14
negative arguments ... 25
NEXT ... 8
next-index-match ... 14
next-line ... 6
next-node ... 11
next-window ... 19
node, selecting from the command line ... 3
nodes, selection of ... 11
numeric arguments ... 24
numeric arguments, negative ... 25

O

O ... 12
online help, using Info as ... 3
options, command line ... 2
outputting to a file ... 3
overriding default key bindings ... 29

P

p ... 11
PageDown ... 8
PageUp ... 9
prev-line ... 6
prev-node ... 11
prev-window ... 19
PREVIOUS ... 8
print-node ... 23
printing ... 23
printing characters, in the echo area ... 21
printing nodes to the local printer ... 23

Q

q ... 25
quit ... 25
quitting ... 25

R

r ... 17
redraw-display ... 10
relative Info file names ... 2
remembering user keystrokes ... 2
repeated search ... 14
replaying recorded keystrokes ... 4
RET, in Info windows ... 17
RET, in the echo area ... 20
RET, vi-like operation ... 9
RIGHT (an arrow key) ... 6
RIGHT, in the echo area ... 20

S

s ... 14
S ... 14
screen, changing the height of ... 25
scroll-backward ... 8
scroll-backward-page-only ... 8
scroll-backward-page-only-set-window ... 9
scroll-behavior ... 27
scroll-forward ... 8
scroll-forward-page-only ... 8
scroll-forward-page-only-set-window ... 8
scroll-half-screen-down ... 9
scroll-half-screen-up ... 9
scroll-other-window ... 19
scroll-step ... 27
scrolling ... 8
scrolling through node structure ... 9
search ... 14
search, and case-sensitivity ... 15
search, case-sensitive ... 14
search-backward ... 14
search-case-sensitively ... 14
search-next ... 14
search-previous ... 14
searching ... 14
Searching all indices ... 2
searching, in the indices ... 14
select-reference-this-line ... 17
select-visited-node ... 13
set-screen-height ... 25
set-variable ... 26
Shift-TAB, in Info windows ... 17
Shift-TAB, in the echo area ... 21
show-footnotes ... 25
show-index-match ... 27
SPC, in Info windows ... 8
SPC, in the echo area ... 21
speech synthesizers ... 4
split-window ... 19

T

t ... 11
TAB, in Info windows ... 17
TAB, in the echo area ... 21
tile-windows ... 19
tiling ... 19
toggle-wrap ... 10
top-node ... 11

U

u ... 11
u, vi-like operation ... 9
universal-argument ... 24
UP (an arrow key) ... 6
UP, vi-like operation ... 9
up-line ... 9
up-node ... 11

V

variables, describing ... 26
variables, setting ... 26
version information ... 4
vi-like key bindings ... 4
view-file ... 13
visible-bell ... 28

W

w, vi-like operation ... 9
where-is ... 24
windows, creating ... 19
windows, deleting ... 19
windows, manipulating ... 18
windows, selecting ... 19

X

xref-item ... 17

Y

y, vi-like operation ... 9

Z

z, vi-like operation ... 8
ZZ, vi-like operation ... 25